AI Chess Assistant - Final Report

Project Option: Incorporate ML Model in a New Use Case Setting

VIVEK VISWAM R. V., SHEETHAL URS, and SAI MOKSHITH GOUD PALLE, University at Buffalo, USA

Abstract
Chess is one of the most played games worldwide. And, it is one of the games that requires practice and consistent learning to be good at it; having a good tutor is one of the best ways to learn chess. Not everyone will have the money afford private tutoring or time to attend a school to learn. There are several online sources that teaches you chess through lessons and puzzles. But such websites don't provide tutoring services, human or AI, so that the students will be able to discuss their strategies during the game. In this project we've developed a software with AI based chat bot integration that lets the users discuss their moves while playing the game. The software also includes Stockfish chess engine integration; it is one of the best chess engines as of date. Based on our experimental queries, we discovered chat bot was able to predict the next best move which was the same move predicted by the Stockfish engine. The chat bot was also able to discuss the game strategies with clarity. Therefore, we are confident that this project is production ready and can be used for real-life use.

1 Motivation and Objective

Chess is one of the games that players can become experts with dedicated practice[Campitelli and Gobet 2008]. There are online platforms such as *Chess.com*[Chess.com Developers 2025] that lets it's users play against each other online. Also, such sites offers lessons to help people learn chess. But doesn't offer live chat bot to discuss the moves while the player is learning.

This concurs with the 2020 study, [Dong and Miao 2020], which clearly states *"The results show that chess online educational products were rich in content and full featured, which could be divided into four categories: overall ecology, video tutorial, tactical training, live broadcast product.* ***However, products still need to improve in product positioning and user experience to promote the development of chess online education."***

Based on the above papers and our own study, existing online chess tutoring services use the following methods to teach:

- **Puzzles**: Daily and weekly puzzles with different chess board configurations to simulate various scenarios.
- **Lessons**: On different chess openings, midgame and endgames strategies.
- **Play with computer**: Offers game with non-human players with option to select the difficulty level.
- **Discussion forums**: People are encouraged to discuss their games and strategies with other users.
- **Online private tutoring**: Chess tutors post their availability on sites such as Fiverr[Fiverr 2025].

Even with all this, we discovered the need to have live tutor that's flexible, time and money-wise, to help students learn chess.

Therefore, by this project we aim to develop an AI chess assistant that will help the users learn chess by helping them understand game strategy and best possible move using a chess engine and an AI powered chat bot.

Our objective includes implementing Nibbler[git [n. d.]b] an open-source chess UI along with Stockfish[Stockfish Developers 2025] chess engine to get the best possible move. And add a chat bot alongside the chessboard, powered by Llama[Grattafiori et al. 2024] - Large Language Model(LLM), so that the user can discuss and learn move the best move and understand the game strategy better. In order to use the LLM within the chess context, we planned to fine-tune it with the [Mitchell J 2025] dataset so that the model will be very familiar with winning moves. And finally use Ollama[git [n. d.]a] an open-source project to host the model server.

2 Existing Work

In [Hassabis 2017], Demis Hassabis reflects on Garry Kasparov's famous 1997 defeat to IBM's Deep Blue and its lasting impact on the field of artificial intelligence. Reviewing Kasparov's book Deep Thinking, Hassabis highlighted how the match was a turning point, showing both the potential and limitations of machine intelligence. He emphasized that the real future of AI lied in collaboration - creating systems that can work with humans.

There were several studies exploring AI-driven chess teaching systems, AI driven approaches, and chess engine advancements. [Silver et al. 2017] introduced AlphaZero, which demonstrated superhuman performance in chess using reinforcement learning without human intervention. [Schrittwieser et al. 2020] worked on that work by mastering multiple games without prior knowledge of the rules. However, both AlphaZero and MuZero are designed for optimal play, not for teaching or explaining moves the best moves. Our approach combines Stockfish[Stockfish Developers 2025] with an AI-powered chat bot to explain moves in real-time as per user's demand.

[Campitelli and Gobet 2008] studied the role of practice in chess expertise, highlighting cognitive and psychological factors in learning the skill. However, their study does not explore AI-driven tools for tutoring. Similarly, [Gobet and Campitelli 2006] examined the influence of memory and practice in chess learning, but their work does not provide interactive AI-powered chat bot. Our project aims to address this by using a AI-powered chat bot.

[McIlroy-Young et al. 2020], a chess AI trained on human games, predicted moves that human players would most likely make. However, it lacked interactive explanations or the ability to engage with users. [Sadikov et al. 2006] developed an intelligent teaching system for teaching chess endgames, but their system was restricted only to specific scenarios. Our project overcomes these limitations by offering interactive move analysis in all phases of the game.

[Touvron et al. 2023] introduced LLaVA, a large multimodal language model that enables advanced visual-linguistic reasoning. [Liu et al. 2023] further improved this model for multimodal tasks, making it highly adaptable for chess-related applications. However,

Authors' Contact Information: Vivek Viswam R. V., vrajabha@buffalo.edu; Sheethal Urs, sheethal@buffalo.edu; Sai Mokshith Goud Palle, saimoksh@buffalo.edu, University at Buffalo, Buffalo, New York, USA.

LLaVA[Liu et al. 2023] has not yet been extensively applied to chess learning. Our system instead leverages Llama[Grattafiori et al. 2024], the model on which LLaVA[Liu et al. 2023] is based on, to recognize board states and generate explanations by getting the states directly from the code without any images.

Several studies have examined AI-assisted chess learning. [He et al. 2019] proposed a deep-learning-based framework for chess analysis, but it focused on move evaluation rather than tutoring. [Wang et al. 2019] reviewed the evolution of chess engines, emphasizing improvements in the algorithms without addressing their educational applications. [Müller et al. 2009] discussed chess engine tuning techniques, which are valuable for competitive play but not for interactive learning. Our project bridges this gap by making Stockfish[Stockfish Developers 2025] an interactive teaching tool rather than just a strong engine.

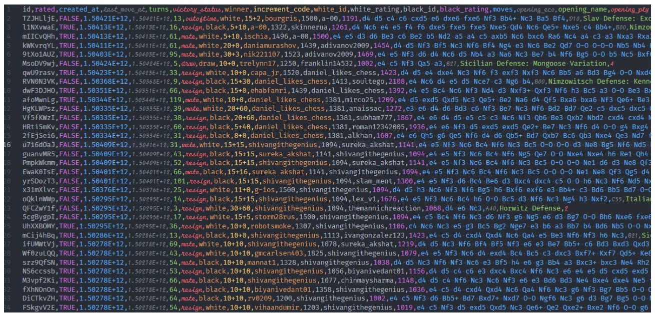
[Tee et al. 2021] examined chat bot-based chess tutoring, showing the potential for interactive AI learning tools. [Knight and Riddle 2018] studied how chess can enhance cognitive and strategic learning, but their work does not use AI for move analysis. [Li and Zhang 2018] introduced Chessbot, a robotic chess-playing system that combined computer vision and AI. However, its primary focus was on gameplay rather than explanation. Our approach uses Llama[Grattafiori et al. 2024] LLM model with a Stockfish[Stockfish Developers 2025] chess engine, ensuring both high-level gameplay and interactive tutoring.

Despite the progress in chess AI, a gap remains in providing real-time, interactive tutoring that combines the analytical strength of chess engines with natural language explanations. Our project seeks to address this by using Stockfish[Stockfish Developers 2025] with Llama[Grattafiori et al. 2024], allowing users to discuss their moves and game strategy.

3 Dataset

The project used [Mitchell J 2025] dataset to fine-tune the Llama[Grattafiori et al. 2024] model. The dataset contains 20,058 past games stored as a *csv* with 16 columns as mentioned below.

- **id**: A unique identifier for each game.
- **rated**: A boolean value indicating whether the game was rated (`True`) or casual (`False`).
- **created_at**: The timestamp indicating when the game was created.
- **last_move_at**: The timestamp of the last recorded move in the game.
- **turns**: The total number of moves played in the game.
- **victory_status**: The method by which the game ended (e.g., *mate*, *resign*, *timeout*).
- **winner**: Specifies whether the winner was *white* or *black*.
- **increment_code**: The time control format used in the game (e.g., `"5+10"` means 5 minutes with a 10-second increment).
- **white_id**: The username or identifier of the player controlling the white pieces.
- **white_rating**: The Elo rating of the white player at the time of the game.
- **black_id**: The username or identifier of the player controlling the black pieces.
- **black_rating**: The Elo rating of the black player at the time of the game.
- **moves**: A sequence of moves played in the game in standard algebraic notation.
- **opening_eco**: The Encyclopedia of Chess Openings (ECO) code associated with the opening played.
- **opening_name**: The descriptive name of the chess opening used in the game.
- **opening_ply**: The number of moves that define the opening phase of the game.



```
id,rated,created_at,last_move_at,turns,victory_status,winner,increment_code,white_id,white_rating,black_id,black_rating,moves,opening_eco,opening_name,opening_ply
TZJHLljE,FALSE,1.50421E+12,1.50421E+12,13,outoftime,white,15+2,bourgris,1500,a-00,1191,d4 d5 c4 c6 cxd5 e6 dxe6 fxe6 Nf3 Bb4+ Nc3 Ba5 Bf4,D10,Slav Defense: Exc
l1NXvwaE,TRUE,1.50413E+12,1.50413E+12,16,resign,black,5+10,a-00,1322,skinnerua,1261,d4 Nc6 e4 e5 f4 f6 dxe5 fxe5 fxe5 Nxe5 Qd4 Nc6 Qe5+ Nxe5 c4 Bb4+,B00,Nimzow
mIICvQHh,TRUE,1.50413E+12,1.50413E+12,61,mate,white,5+10,ischia,1496,a-00,1500,e4 e5 d3 d6 Be3 c6 Be2 b5 Nd2 a5 a4 c5 axb5 Nc6 bxc6 Ra6 Nc4 a4 c3 a3 Nxa3 Rxa
kWKvrqYL,TRUE,1.50411E+12,1.50411E+12,61,mate,white,20+0,daniamurashov,1439,adivanov2009,1454,d4 d5 Nf3 Bf5 Nc3 Nf6 Bf4 Ng4 e3 Nc6 Be2 Qd7 O-O O-O-O Nb5 Nb4
9tXo1AUZ,TRUE,1.50403E+12,1.50403E+12,95,mate,white,30+3,nik221107,1523,adivanov2009,1469,e4 e5 Nf3 d6 d4 Nc6 d5 Nb4 a3 Na6 Nc3 Be7 b4 Nf6 Bg5 O-O b5 Nc5 Bxf
MsoDV9wj,FALSE,1.50424E+12,1.50424E+12,5,draw,draw,10+0,trelynn17,1250,franklin14532,1002,e4 c5 Nf3 Qa5 a3,B27,Sicilian Defense: Mongoose Variation,4
qwU9rasv,TRUE,1.50423E+12,1.50423E+12,33,resign,white,10+0,capa_jr,1520,daniel_likes_chess,1423,d4 d5 e4 dxe4 Nc3 Nf6 f3 exf3 Nxf3 Nc6 Bb5 a6 Bd3 Bg4 O-O Nxd4
RVN0N3VK,FALSE,1.50368E+12,1.50368E+12,9,resign,black,15+30,daniel_likes_chess,1413,soultego,2108,e4 Nc6 d4 e5 d5 Nce7 c3 Ng6 b4,B00,Nimzowitsch Defense: Kenn
dwF3DJHO,TRUE,1.50351E+12,1.50351E+12,66,resign,black,15+0,ehabfanri,1439,daniel_likes_chess,1392,e4 e5 Bc4 Nc6 Nf3 Nd4 d3 Nxf3+ Qxf3 Nf6 h3 Bc5 a3 O-O Be3 Bx
afoMwnLg,TRUE,1.50344E+12,1.50344E+12,119,mate,white,10+0,daniel_likes_chess,1381,mirco25,1209,e4 d5 exd5 Qxd5 Nc3 Qe5+ Be2 Na6 d4 Qf5 Bxa6 bxa6 Nf3 Qe6+ Be3
HgKLWPsz,FALSE,1.50335E+12,1.50335E+12,39,mate,white,20+60,daniel_likes_chess,1381,anaissac,1272,e3 e6 d4 d6 Bd3 c6 Nf3 Be7 Nc3 Nf6 Bd2 Bd7 Qe2 c5 dxc5 dxc5
Vf5fKWzI,FALSE,1.50335E+12,1.50335E+12,38,resign,black,20+60,daniel_likes_chess,1381,subham777,1867,e4 e6 d4 d5 e5 c5 c3 Nc6 Nf3 Qb6 Be3 Qxb2 Nbd2 cxd4 cxd4
HRti5mKv,FALSE,1.50335E+12,1.50335E+12,60,resign,black,5+40,daniel_likes_chess,1381,roman12342005,1936,e4 e6 Nf3 d5 exd5 exd5 Qe2+ Be7 Nc3 Nf6 d4 O-O g4 Bxg4
2fEjSei6,FALSE,1.50334E+12,1.50334E+12,31,resign,black,8+0,daniel_likes_chess,1381,alkhan,1607,e4 e6 Qh5 g6 Qe5 Nf6 d4 d6 Qb5+ Bd7 Qxb7 Bc6 Qb3 Nxe4 Qe3 Nd7
u7i6dOaJ,FALSE,1.50409E+12,1.5041E+12,31,mate,white,15+15,shivangithegenius,1094,sureka_akshat,1141,e4 e5 Nf3 Nc6 Bc4 Nf6 Nc3 Bc5 O-O O-O d3 Ne8 Bg5 Nf6 Nd5
guanvMR5,FALSE,1.50409E+12,1.50409E+12,43,resign,black,15+15,sureka_akshat,1141,shivangithegenius,1094,e4 e5 Nf3 Nc6 Bc4 Nf6 Ng5 Qe7 O-O Nxe4 Nxe4 h6 Re1 Qh4
PmpkWkmm,FALSE,1.50409E+12,1.50409E+12,52,resign,black,15+15,shivangithegenius,1094,sureka_akshat,1141,e4 e5 Nf3 Nc6 Bc4 Nf6 Nc3 Bc5 O-O O-O Ne1 d6 d3 Ne8 Qf3
EwaK0IsE,FALSE,1.50401E+12,1.50401E+12,66,mate,black,15+16,sureka_akshat,1141,shivangithegenius,1094,e4 e5 Nf3 Nc6 Bc4 Nf6 Nc3 Bc5 O-O O-O Ne1 Ne8 Qf3 Qg5 d4
yrSDozT3,FALSE,1.50401E+12,1.50401E+12,101,resign,black,15+15,shivangithegenius,1094,slam_ment,1300,e4 e5 Nf3 d6 Bc4 Be6 d3 Bxc4 dxc4 c5 O-O h6 Nc3 Nf6 Nd5 Nx
x31mXlvc,FALSE,1.50376E+12,1.50376E+12,25,resign,white,11+0,g-ios,1500,shivangithegenius,1094,d4 d5 h3 Nc6 Nf3 Nf6 Bg5 h6 Bxf6 exf6 e3 Bb4+ c3 Bd6 Bb5 Bd7 O-O
oQklnWWp,FALSE,1.50295E+12,1.50295E+12,14,resign,black,15+15,shivangithegenius,1094,lex_v1,1676,e4 e5 Nf3 Nc6 Bc4 h6 O-O Bc5 d3 Nf6 Nc3 Ng4 h3 Nxf2,C55,Italian
QFCZwY1f,FALSE,1.50295E+12,1.50295E+12,3,resign,white,30+60,shivangithegenius,1094,themannichreaction,1068,d4 e6 Nc3,A40,Horwitz Defense,2
5cgBygpI,FALSE,1.50295E+12,1.50295E+12,17,resign,white,15+5,storm28rus,1500,shivangithegenius,1094,e4 c5 Bc4 Nf6 Nc3 d6 Nf3 g6 Ng5 e6 d3 Bg7 O-O Bh6 Nxe6 fxe6
UhXXBOMY,TRUE,1.50295E+12,1.50295E+12,36,resign,white,10+0,robotsmoke,1307,shivangithegenius,1106,c4 Nc6 Nc3 e5 g3 Bc5 Bg2 Nge7 e3 b6 a3 Bb7 b4 Bd6 Nb5 O-O Nx
mCij4hBq,TRUE,1.50286E+12,1.50286E+12,13,resign,black,10+0,shivangithegenius,1113,ivangonzalez123,1423,e4 c5 d4 cxd4 Qxd4 Nc6 Qa4 e5 Be3 Nf6 Nf3 h6 Nc3,B21,Si
ifUMWtVj,TRUE,1.50278E+12,1.50278E+12,69,mate,white,10+10,shivangithegenius,1078,sureka_akshat,1219,d4 d5 Nc3 Nf6 Bf4 Bf5 Nf3 e6 e3 Be7 Bb5+ c6 Bd3 Bxd3 Qxd3
Wf0zuLQQ,TRUE,1.50278E+12,1.50278E+12,43,resign,white,10+10,gmcarlsen403,1825,shivangithegenius,1079,e4 e5 Nf3 Nc6 d4 exd4 Bc4 Bc5 c3 dxc3 Bxf7+ Kxf7 Qd5+ Ke8
srz9QfSN,TRUE,1.50278E+12,1.50278E+12,54,mate,black,10+10,mannat1,1328,shivangithegenius,1038,d4 d5 Nc3 Nf6 Nf3 Nc6 e3 Bf5 h4 e6 g3 Bb4 a3 Bxc3+ bxc3 Ne4 Rh2
NS6ccssb,TRUE,1.50278E+12,1.50278E+12,53,resign,black,10+10,shivangithegenius,1056,biyanivedant01,1156,d4 d5 c4 c6 e3 dxc4 Bxc4 Nf6 Nc3 e6 e4 e5 d5 cxd5 exd5
M3vpf2Ki,TRUE,1.50278E+12,1.50278E+12,66,mate,black,10+10,shivangithegenius,1077,chinmaysharma,1148,d4 d5 c4 Nf6 Nc3 Nc6 Nf3 e6 e3 Bd6 Bd3 Ne4 Bxe4 dxe4 Ne5
fXhNOnOn,TRUE,1.50278E+12,1.50278E+12,64,resign,black,10+10,biyanivedant01,1358,shivangithegenius,1036,e4 c5 d4 cxd4 Qxd4 Nc6 Qa4 Nf6 Nc3 g6 Nf3 Bg7 Bb5 O-O
DiCTkvZH,TRUE,1.50278E+12,1.50278E+12,64,mate,black,10+10,rv0209,1200,shivangithegenius,1002,e4 c5 Nf3 d6 Bb5+ Bd7 Bxd7+ Nxd7 O-O Ngf6 Nc3 g6 d3 Bg7 Bg5 O-O
FSkgvV2E,TRUE,1.50278E+12,1.50278E+12,54,resign,white,10+10,vihaandumir,1203,shivangithegenius,1019,e4 c5 Nf3 d5 exd5 Qxd5 Nc3 Qe6+ Qe2 Qxe2+ Bxe2 Nf6 O-O g6
```

Fig. 1. Original dataset

3.1 Original Dataset Statistics

The following are the statistics on the various variables in the original dataset.

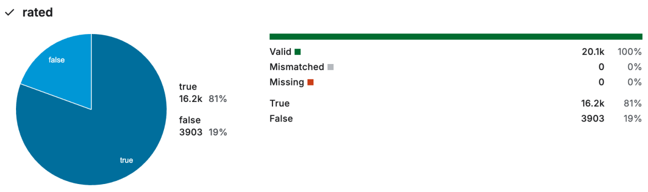


Fig. 2. Rating statistics

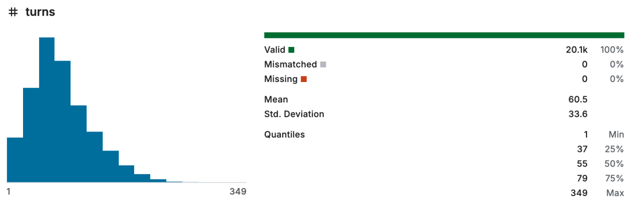
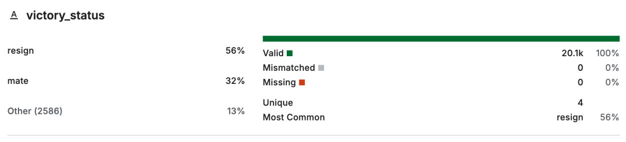


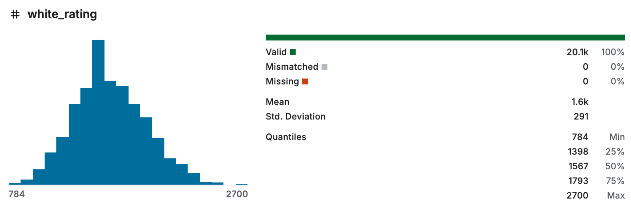
Fig. 3. Turns statistics



victory_status

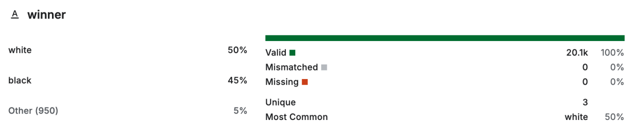
resign	56%	Valid 🟩	20.1k	100%
		Mismatched	0	0%
mate	32%	Missing 🟥	0	0%
Other (2586)	13%	Unique	4	
		Most Common	resign	56%

Fig. 4. Victory status statistics



Valid 🟩	20.1k	100%
Mismatched	0	0%
Missing 🟥	0	0%
Mean	1.6k	
Std. Deviation	291	
Quantiles	784	Min
	1398	25%
	1567	50%
	1793	75%
	2700	Max

Fig. 5. White-rating statistics



winner

white	50%	Valid 🟩	20.1k	100%
		Mismatched	0	0%
black	45%	Missing 🟥	0	0%
Other (950)	5%	Unique	3	
		Most Common	white	50%

Fig. 6. Winner statistics

3.2 Modified Dataset

The dataset contained columns irrelevant to the fine-tuning of our LLM. Columns such as *id, rated, created*$_a$*twhicharenon−exhaustive, wereirrelevant.Inaddition, fine−tuningourmodelrequiresustostructureitasalistofpromptsandresponses.Therefore, wewroteaPythonscripttocreateamodifieddataset.HereisthePythonscri*

Selecting Relevant Columns

- *Only the columns essential for prompt and response generation are retained:*
  - `victory_status`, `winner`, `white_rating`, `black_rati` `moves`, *and* `opening_name`.
- *This step extracts the data most useful for generating descriptive human-like narratives about the games.*

Defining the Prompt-Response Generator

- *A function named* `format_humanized_prompt` *was defined, which took a row of the dataset and returned a dictionary with two keys:* `prompt` *and* `response`.
- *The prompt was designed to resemble a user's natural-language query about the chess game. It included questions like:*
  - *"What is the name of the opening?"*
  - *"Who won?"*
  - *"How did the winner win?"*
  - *"Can you predict the ratings of the players?"*
- *The game's moves were appended to these questions for context.*

Generating a Human-Like Response

- *The function also constructed a response that sounds like a natural reply from an assistant:*
  - *The name of the opening was referenced.*
  - *The winner was mentioned.*
  - *The method of victory was clarified — e.g., "outoftime" was replaced with "timeout".*
  - *An estimate of the players' ratings were included to give the model a better understanding of the players.*

Transforming the Dataset

- *The* `format_humanized_prompt` *function was applied to each row of the DataFrame using* `df.apply(..., axis=1)`.
- *The result was a list of dictionaries each with a* `prompt` *and* `response` *suitable for instruction tuning.*

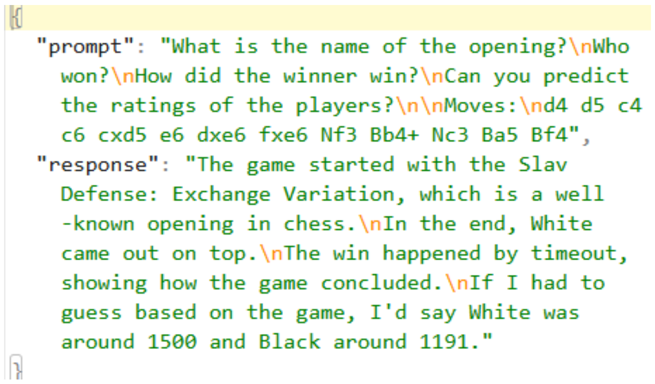
Saving to JSONL Format

- *The generated prompt-response pairs were saved to a file named* `chess_humanized_prompts.jsonl`.
- *Each dictionary is serialized to a JSON string and written line-by-line, which is the standard format for large-scale language model fine-tuning datasets.*
- `ensure_ascii=False` *was used to preserve any special characters.*

We chose the following headers to build our new dataset as we felt that these were the most relevant ones to be learned:

- **victory_status**: The method by which the game ended (e.g., *mate, resign, timeout*).
- **winner**: Specifies whether the winner was *white* or *black*.
- **white_rating**: The Elo rating of the white player at the time of the game.
- **black_rating**: The Elo rating of the black player at the time of the game.
- **moves**: A sequence of moves played in the game in standard algebraic notation.
- **opening_name**: The descriptive name of the chess opening used in the game.

Fig. 7. Modified dataset



```
{
  "prompt": "What is the name of the opening?\nWho
    won?\nHow did the winner win?\nCan you predict
    the ratings of the players?\n\nMoves:\nd4 d5 c4
    c6 cxd5 e6 dxe6 fxe6 Nf3 Bb4+ Nc3 Ba5 Bf4",
  "response": "The game started with the Slav
    Defense: Exchange Variation, which is a well
    -known opening in chess.\nIn the end, White
    came out on top.\nThe win happened by timeout,
    showing how the game concluded.\nIf I had to
    guess based on the game, I'd say White was
    around 1500 and Black around 1191."
}
```

Fig. 8. Single datapoint from new dataset

4 Our Approach

After a thorough analysis we designed our project to have three different components:

- Chess Engine
- Chess UI
- AI powered chat bot

We selected our individual components based on the following analysis.

4.1 Chess Engine

We explored several open-source chess engines, each with different strengths and functionalities. Leela Chess Zero (Lc0)[Developers 2023f] was a neural network-based engine inspired by AlphaZero, it utilized Monte Carlo Tree Search(MCTS) for move selection. It learned through self-play, making it highly adaptive but it required significant computational resources, particularly a GPU, for good performance. Berserk[Developers 2023b] balanced classical evaluation and neural network-based decision-making, offering fast and efficient response. Similarly, Ethereal[Grant 2023] focused on custom evaluation functions and traditional heuristics, making it efficient but less adaptable to modern AI-driven approaches.

For chess variants, Fairy-Stockfish[Developers 2023d] extended Stockfish's capabilities to support non-standard formats like Crazyhouse and Fischer Random, making it a preferred choice for variant chess players. Schooner[Dugovic 2023] was a Rust-based multi-threaded engine designed for AI research and experimentation. Marvin[Developers 2023g] was another strong open-source engine, incorporating advanced pruning techniques to optimize search depth and improve efficiency. Texel[Trevithick 2023] emphasized positional play, using a tuned evaluation function to refine decision-making, making it a strong candidate for strategic chess analysis.

But among these engines, Stockfish[Stockfish Developers 2025] was the most powerful and widely adopted chess engine. It used advanced alpha-beta pruning, Efficiently Updatable Neural Networks, and a highly optimized search algorithm. Also, Stockfish[Stockfish Developers 2025] doesn't use GPU acceleration and it had good Universal Chess Interface(UCI) support. Therefore, we chose Stockfish[Stockfish Developers 2025] as our chess engine for our AI chess assistant.

4.2 Chess UI

We explored several open-source chess UIs. Arena Chess GUI [Developers 2023a] was a widely used interface that supported UCI and WinBoard protocols. It provided features like engine tournaments and analysis tools, making it suitable for both casual and advanced users. However, its interface was outdated.

Scid vs. PC [vs PC Developers 2023] was another powerful GUI, known for its extensive database management features. It allowed users to maintain and analyze large collections of games and offered player statistics and opening preparation tools too. But its interface was outdated and required a learning curve for us. Cute Chess [Developers 2023c] had a clean and easy to use interface. However, it lacked advanced database features found in other UI frameworks.

Lucas Chess [Developers 2023e] was interface. It provides a variety of training modes and lessons, but it wasn't feature-rich for advanced engine analysis in Stockfish[Stockfish Developers 2025].

Among these options, Nibbler [Developers 2023h] was the best choice for our project. It was a simple and easy to use framework. And, considering our experience we felt that this was easier to work with. It was a Node[nod [n. d.]] application so, knowing JavaScript was enough to modify the code per our requirements.

4.3 AI Powered Chat Bot

We compared a number of Large Language Models(LLMs) before deciding on Llama 3.1 to build an AI-driven chatbot that helps users to understand game strategy. We also considered the Claude 3, Mistral, and GPT-4 models.

GPT-4 [OpenAI 2023] from OpenAI was the most powerful proprietary LLM, due to its advanced reasoning capabilities and its extensive knowledge base. But it had its limitations since we could not fine-tune the model for chess-specific tasks due to its closed-source nature. Moreover, the incorporation of GPT-4 required interactions via API, which proved to be expensive.

Mistral [AI 2023b] was another alternative open-source, provides extraordinary efficiency and strong reasoning abilities. Although it offered a good balance of performance and computational cost, it did not have the extensive fine-tuning support or specialized instruction-following capabilities necessary for detailed chess strategy explanations.

Claude 3 [AI 2023a] by Anthropic was designed, focused mainly on conversational AI. Although it is effective at determining user requirements and offered thorough justifications, it was still a proprietary model, which made offline integration and customization impossible for our use case.

Llama 3.1 [Grattafiori et al. 2024] was the best option for our chess assistance chat bot after a thorough comparison. We may modify Llama for chess-specific insights because it is completely open-source, in contrast to proprietary models like GPT-4 and Claude 3. Llama had better instruction-following capabilities than Mistral and Falcon after comparing its performance through our interactions, which made it more useful for assisting players in analyzing intricate game strategies.

Thus, Llama 3.1 was the best option for our AI-powered chess assistant as it combined efficiency, advanced reasoning capabilities, and customizability.

The project used [Mitchell J 2025] dataset to fine-tune the Llama[Grattafiori et al. 2024] model.

At this point, we've set up Stockfish[Stockfish Developers 2025] along with Nibbler[Developers 2023h] in our local machine. The next step is to create a chat bot section on Nibbler[Developers 2023h] UI and connect the Llama[Grattafiori et al. 2024] model, fine-tuned with [Mitchell J 2025] dataset.

Users running Nibbler[Developers 2023h] will connect with the Ollama[git [n. d.]a] server to send and receive messages with the Llama[Grattafiori et al. 2024] model while playing the game.

4.4 Fine-tuning

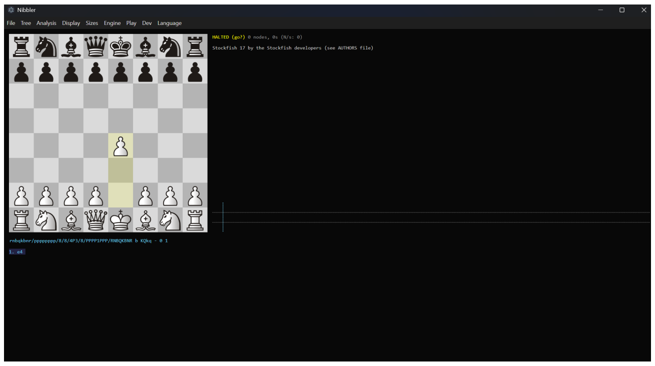
Before using the Llama[Grattafiori et al. 2024] model we had to fine-tune the model so that it will be able to answer chess related queries. The dataset we chose was not designed to be used to fine-tune LLMs. Therefore, we created a modified dataset using a Python script as mentioned before. So, the model has a better understanding of the past chess games to answer the queries. We also discovered that the model without fine-tuning gave a satisfactory performance, yet we went ahead with the fine-tuning process.

We followed the following steps to fine-tune the model:

- **Model and Tokenizer Initialization**
  - Loaded the LLaMA 3.1 8B-Instruct model and tokenizer using the Hugging Face Transformers library.
  - Enabled fast tokenization with `use_fast=True` for improved performance.
  - Loaded the model in 4-bit precision (`load_in_4bit=True`) and used automatic device mapping (`device_map="auto"`) to distribute the model across all available GPU devices.
  - This approach significantly reduces the VRAM required.
- **Dataset Loading and Preprocessing**
  - Loaded a JSON Lines dataset file (`chess_humanized_prompts.jsonl`) that contains chess-related prompt-response pairs.
  - Split the dataset into training and test sets with a 90/10 ratio using `train_test_split()`.
  - Set the tokenizer's padding token to be the same as its end-of-sequence token to avoid errors during padding.
  - Defined a custom `format()` function that wraps each prompt-response pair in special tokens used by the LLaMA instruction format:
    - Each example was structured with special markers like `<|begin_of_text|>`, `<|start_header_id|>user <|end_header_id|>`, and `<|eot_id|>` to tell the model how to treat the text.
  - The formatted text was then tokenized using the tokenizer, with a maximum sequence length of 2048 and padding to that maximum length.
- **Parameter-Efficient Fine-Tuning (PEFT) with LoRA**
  - Used LoRA (Low-Rank Adaptation) to fine-tune only a small subset of parameters, greatly reducing memory requirements and training time.
  - Configured the `LoraConfig` with:
    - Rank $r = 8$, `lora_alpha = 16`, and `lora_dropout = 0.05`
    - Targeted attention projection layers: `q_proj`, `k_proj`, `v_proj`, `o_proj`
  - The model was then wrapped with `get_peft_model()` to apply the LoRA modifications.
  - Printed the number of trainable parameters to verify that only a small fraction of the model is being updated during training.
- **Training Configuration and Execution**
  - Created a set of training arguments specifying:
    - Batch size of 1, gradient accumulation over 4 steps to simulate a batch size of 4.
    - 1 epoch of training, with a learning rate of 2×10^{-4}.
    - Mixed precision training with `fp16=True` for improved speed and lower memory usage.
    - Logging every 25 steps and saving the model at the end of each epoch.
  - Initialized a `Trainer` object, passing in:
    - The PEFT model and tokenizer
    - The training and evaluation datasets
    - A data collator for language modeling with `mlm=False`, meaning causal (not masked) language modeling is used.
  - Started training using the `train()` method of the trainer.
- **Saving the Fine-Tuned Model**
  - Saved both the model and tokenizer to the local directory `llama3-chess-finetuned`.
  - This step ensures that the trained weights and tokenizer can be reused later or loaded for inference.
- **Merging LoRA Adapters and Uploading to Hugging Face**
  - Reloaded the base LLaMA 3 model and loaded the fine-tuned adapter weights using `PeftModel.from_pretrained()`.
  - Called `merge_and_unload()` to combine the LoRA adapter with the base model into a single unified model.
  - Pushed the merged model and tokenizer to the Hugging Face Hub using `push_to_hub()` for public or private sharing.

5 Results

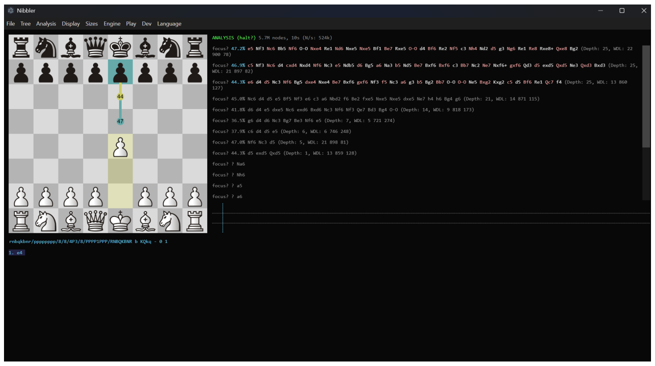
As mentioned in the above section, we've set up Nibbler[Developers 2023h] along with Stockfish[Stockfish Developers 2025] engine. The following are the screenshots of the set up where two human players play each other in our environment. In the figure 9, you



```
HALTED (go?) 0 nodes, 0s (N/s: 0)
Stockfish 17 by the Stockfish developers (see AUTHORS file)
rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq - 0 1
1. e4
```

Fig. 9. Without Stockfish

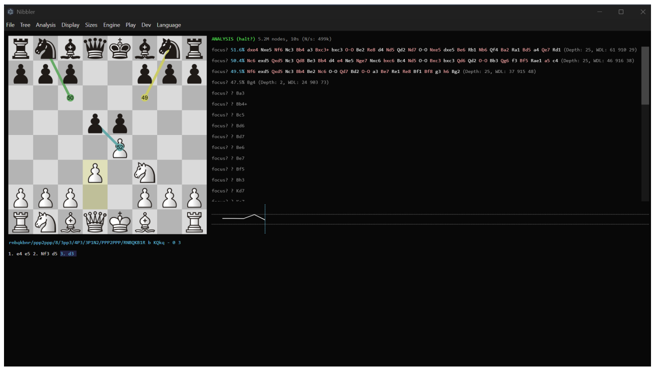
could see that there aren't any analytics on the right hand side of the chessboard or on the board itself. But after integrating Stockfish engine, you could see in figure 10 that the an arrow appears on the



```
ANALYSIS (halt?) 5.7M nodes, 10s (N/s: 524k)
focus? 47.2% e5 Nf3 Nc6 Bb5 Nf6 O-O Nxe4 Re1 Nd6 Nxe5 Nxe5 Bf1 Be7 Rxe5 O-O d4 Bf6 Re2 Nf5 c3 Nh4 Nd2 d5 g3 Ng6 Re1 Re8 Rxe8+ Qxe8 Bg2 (Depth: 25, WDL: 22 900 78)
focus? 46.9% c5 Nf3 Nc6 d4 cxd4 Nxd4 Nf6 Nc3 e5 Ndb5 d6 Bg5 a6 Na3 b5 Nd5 Be7 Bxf6 Bxf6 c3 Bb7 Nc2 Ne7 Nxf6+ gxf6 Qd3 d5 exd5 Qxd5 Ne3 Qxd3 Bxd3 (Depth: 25, WDL: 21 897 82)
focus? 44.3% e6 d4 d5 Nc3 Nf6 Bg5 dxe4 Nxe4 Be7 Bxf6 gxf6 Nf3 f5 Nc3 a6 g3 b5 Bg2 Bb7 O-O O-O Ne5 Bxg2 Kxg2 c5 d5 Bf6 Re1 Qc7 f4 (Depth: 25, WDL: 13 860 127)
focus? 45.0% Nc6 d4 d5 e5 Bf5 Nf3 e6 c3 a6 Nbd2 f6 Be2 fxe5 Nxe5 Nxe5 dxe5 Ne7 h4 h6 Bg4 g6 (Depth: 21, WDL: 14 871 115)
focus? 41.8% d6 d4 e5 dxe5 Nc6 exd6 Bxd6 Nc3 Nf6 Nf3 Qe7 Bd3 Bg4 O-O (Depth: 14, WDL: 9 818 173)
focus? 36.5% g6 d4 d6 Nc3 Bg7 Be3 Nf6 e5 (Depth: 7, WDL: 5 721 274)
focus? 37.9% c6 d4 d5 e5 (Depth: 6, WDL: 6 746 248)
focus? 47.0% Nf6 Nc3 d5 (Depth: 5, WDL: 21 898 81)
focus? 44.3% d5 exd5 Qxd5 (Depth: 1, WDL: 13 859 128)
focus? ? Na6
focus? ? Nh6
focus? ? a5
focus? ? a6

rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq - 0 1
1. e4
```

Fig. 10. First Move With Stockfish

board along with the score computed by the Stockfish engine along with the computation of the score on the right hand side of the board. As you can see in figure 11, the multiple moves are suggested



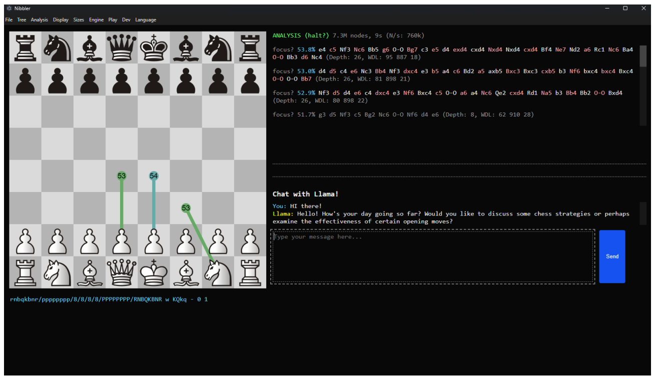
```
ANALYSIS (halt?) 5.2M nodes, 10s (N/s: 499k)
focus? 51.6% dxe4 Nxe5 Nf6 Nc3 Bb4 a3 Bxc3+ bxc3 O-O Be2 Re8 d4 Nd5 Qd2 Nd7 O-O Nxe5 dxe5 Be6 Rb1 Nb6 Qf4 Ba2 Ra1 Bd5 a4 Qe7 Rd1 (Depth: 25, WDL: 61 910 29)
focus? 50.4% Nc6 exd5 Qxd5 Nc3 Qd8 Be3 Bb4 d4 e4 Ne5 Nge7 Nxc6 bxc6 Bc4 Nd5 O-O Bxc3 bxc3 Qd6 Qd2 O-O Bb3 Qg6 f3 Bf5 Rae1 a5 c4 (Depth: 25, WDL: 46 916 38)
focus? 49.5% Nf6 exd5 Qxd5 Nc3 Bb4 Be2 Nc6 O-O Qd7 Bd2 O-O a3 Be7 Re1 Re8 Bf1 Bf8 g3 h6 Bg2 (Depth: 25, WDL: 37 915 48)
focus? 47.5% Bg4 (Depth: 2, WDL: 24 903 73)
focus? ? Ba3
focus? ? Bb4+
focus? ? Bc5
focus? ? Bd6
focus? ? Bd7
focus? ? Be6
focus? ? Be7
focus? ? Bf5
focus? ? Bh3
focus? ? Kd7

rnbqkbnr/ppp2ppp/8/3pp3/4P3/3P1N2/PPP2PPP/RNBQKB1R b KQkq - 0 3
1. e4 e5 2. Nf3 d5 3. d3
```

Fig. 11. Game Progession

with their Stockfish score. The user could choose the move with the highest score. After integrating the chat bot to the right-hand side we performed a few queries. Our goal was to test the model performance on the following tasks:

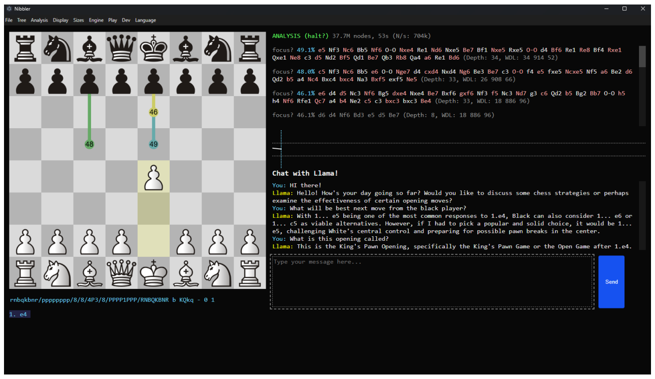
(1) **Greetings**: Determine how the model will handle greetings by the player. Refer to figure 12.
(2) **Understanding of opening moves**: We tested this by querying the model the name of the opening that we made as a white player. Refer to figure 13.
(3) **Game strategy(determining next best move)**: We wanted to test how the model will predict the next best move and provide an explanation. Therefore, we queried the best move as a white player. Refer to figure 14.
(4) **Ability to understand opponent's move**: We asked the model what move might the opponent make based on our latest move. Refer to 15.
(5) **Handling non-chess queries**: We asked the model a non-chess related question to determine how well the model was able to handle off-topic queries. Which was important to keep the focus on the game. Refer to figure 16.

The model performed exceptionally well on all our test queries and proved to be a good model to handling various possible scenarios.



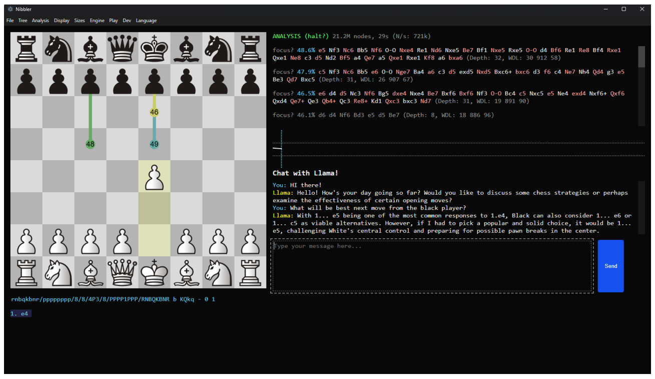
```
ANALYSIS (halt?) 7.3M nodes, 9s (N/s: 760k)
focus? 53.8% e4 c5 Nf3 Nc6 Bb5 g6 O-O Bg7 c3 e5 d4 exd4 cxd4 Nxd4 Nxd4 cxd4 Bf4 Ne7 Nd2 a6 Rc1 Nc6 Ba4 O-O Bb3 d6 Nc4 (Depth: 26, WDL: 95 887 18)
focus? 53.0% d4 d5 c4 e6 Nc3 Bb4 Nf3 dxc4 e3 b5 a4 c6 Bd2 a5 axb5 Bxc3 Bxc3 cxb5 b3 Nf6 bxc4 bxc4 Bxc4 O-O O-O Bb7 (Depth: 26, WDL: 81 898 21)
focus? 52.9% Nf3 d5 d4 e6 c4 dxc4 e3 Nf6 Bxc4 c5 O-O a6 a4 Nc6 Qe2 cxd4 Rd1 Na5 b3 Bb4 Bb2 O-O Bxd4 (Depth: 26, WDL: 80 898 22)
focus? 51.7% g3 d5 Nf3 c5 Bg2 Nc6 O-O Nf6 d4 e6 (Depth: 8, WDL: 62 910 28)

Chat with Llama!
You: HI there!
Llama: Hello! How's your day going so far? Would you like to discuss some chess strategies or perhaps examine the effectiveness of certain opening moves?
Type your message here...

rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
```

Fig. 12. Greetings



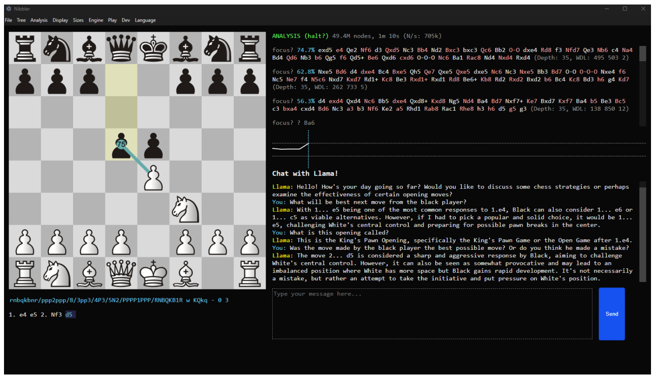
```
ANALYSIS (halt?) 37.7M nodes, 53s (N/s: 704k)
focus? 49.1% e5 Nf3 Nc6 Bb5 Nf6 O-O Nxe4 Re1 Nd6 Nxe5 Be7 Bf1 Nxe5 Rxe5 O-O d4 Bf6 Re1 Re8 Bf4 Rxe1 Qxe1 Ne8 c3 d5 Nd2 Bf5 Qd1 Be7 Qb3 Rb8 Qa4 a6 Re1 Bd6 (Depth: 34, WDL: 34 914 52)
focus? 48.0% c5 Nf3 Nc6 Bb5 e6 O-O Nge7 d4 cxd4 Nxd4 Ng6 Be3 Be7 c3 O-O f4 e5 fxe5 Ncxe5 Nf5 a6 Be2 d6 Qd2 b5 a4 Nc4 Bxc4 bxc4 Na3 Bxf5 exf5 Ne5 (Depth: 33, WDL: 26 908 66)
focus? 46.1% e6 d4 d5 Nc3 Nf6 Bg5 dxe4 Nxe4 Be7 Bxf6 gxf6 Nf3 f5 Nc3 Nd7 g3 c6 Qd2 b5 Bg2 Bb7 O-O h5 h4 Nf6 Rfe1 Qc7 a4 b4 Ne2 c5 c3 bxc3 bxc3 Be4 (Depth: 33, WDL: 18 886 96)
focus? 46.1% d6 d4 Nf6 Bd3 e5 d5 Be7 (Depth: 8, WDL: 18 886 96)

Chat with Llama!
You: HI there!
Llama: Hello! How's your day going so far? Would you like to discuss some chess strategies or perhaps examine the effectiveness of certain opening moves?
You: What will be best next move from the black player?
Llama: With 1... e5 being one of the most common responses to 1.e4, Black can also consider 1... e6 or 1... c5 as viable alternatives. However, if I had to pick a popular and solid choice, it would be 1... e5, challenging White's central control and preparing for possible pawn breaks in the center.
You: What is this opening called?
Llama: This is the King's Pawn Opening, specifically the King's Pawn Game or the Open Game after 1.e4.
Type your message here...

rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq - 0 1
1. e4
```

Fig. 13. Llama explaining opening move



```
ANALYSIS (halt?) 21.2M nodes, 29s (N/s: 721k)
focus? 48.6% e5 Nf3 Nc6 Bb5 Nf6 O-O Nxe4 Re1 Nd6 Nxe5 Be7 Bf1 Nxe5 Rxe5 O-O d4 Bf6 Re1 Re8 Bf4 Rxe1 Qxe1 Ne8 c3 d5 Nd2 Bf5 a4 Qe7 a5 Qxe1 Rxe1 Kf8 a6 bxa6 (Depth: 32, WDL: 30 912 58)
focus? 47.9% c5 Nf3 Nc6 Bb5 e6 O-O Nge7 Ba4 a6 c3 d5 exd5 Nxd5 Bxc6+ bxc6 d3 f6 c4 Ne7 Nh4 Qd4 g3 e5 Be3 Qd7 Bxc5 (Depth: 31, WDL: 26 907 67)
focus? 46.5% e6 d4 d5 Nc3 Nf6 Bg5 dxe4 Nxe4 Be7 Bxf6 Bxf6 Nf3 O-O Bc4 c5 Nxc5 e5 Ne4 exd4 Nxf6+ Qxf6 Qxd4 Qe7+ Qe3 Qb4+ Qc3 Re8+ Kd1 Qxc3 bxc3 Nd7 (Depth: 31, WDL: 19 891 90)
focus? 46.1% d6 d4 Nf6 Bd3 e5 d5 Be7 (Depth: 8, WDL: 18 886 96)

Chat with Llama!
You: HI there!
Llama: Hello! How's your day going so far? Would you like to discuss some chess strategies or perhaps examine the effectiveness of certain opening moves?
You: What will be best next move from the black player?
Llama: With 1... e5 being one of the most common responses to 1.e4, Black can also consider 1... e6 or 1... c5 as viable alternatives. However, if I had to pick a popular and solid choice, it would be 1... e5, challenging White's central control and preparing for possible pawn breaks in the center.
Type your message here...

rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq - 0 1
1. e4
```

Fig. 14. Game strategy



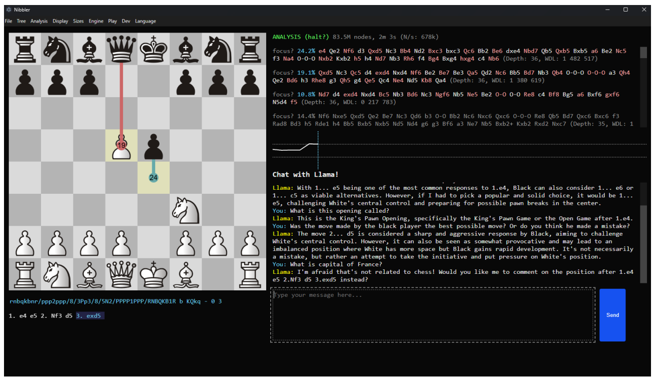
```
ANALYSIS (halt?) 49.4M nodes, 1m 10s (N/s: 705k)

focus? 74.7% exd5 e4 Qe2 Nf6 d3 Qxd5 Nc3 Bb4 Nd2 Bxc3 bxc3 Qc6 Bb2 O-O dxe4 Rd8 f3 Nfd7 Qe3 Nb6 c4 Na4
Bd4 Qd6 Nb3 b6 Qg5 f6 Qd5+ Be6 Qxd6 cxd6 O-O-O Nc6 Ba1 Rac8 Nd4 Nxd4 Rxd4 (Depth: 35, WDL: 495 503 2)

focus? 62.8% Nxe5 Bd6 d4 dxe4 Bc4 Bxe5 Qh5 Qe7 Qxe5 Qxe5 dxe5 Nc6 Nc3 Nxe5 Bb3 Bd7 O-O O-O-O Nxe4 f6
Nc5 Ne7 f4 N5c6 Nxd7 Kxd7 Rd1+ Kc8 Be3 Rxd1+ Rxd1 Rd8 Be6+ Kb8 Rd2 Rxd2 Bxd2 b6 Bc4 Kc8 Bd3 h6 g4 Kd7
(Depth: 35, WDL: 262 733 5)

focus? 56.3% d4 exd4 Qxd4 Nc6 Bb5 dxe4 Qxd8+ Kxd8 Ng5 Nd4 Ba4 Bd7 Nxf7+ Ke7 Bxd7 Kxf7 Ba4 b5 Be3 Bc5
c3 bxa4 cxd4 Bd6 Nc3 a3 b3 Nf6 Ke2 a5 Rhd1 Rab8 Rac1 Rhe8 h3 h6 d5 g5 g3 (Depth: 35, WDL: 138 850 12)

focus? ? Ba6
```

Chat with Llama!

```
Llama: Hello! How's your day going so far? Would you like to discuss some chess strategies or perhaps
examine the effectiveness of certain opening moves?
You: What will be best next move from the black player?
Llama: With 1... e5 being one of the most common responses to 1.e4, Black can also consider 1... e6 or
1... c5 as viable alternatives. However, if I had to pick a popular and solid choice, it would be 1...
e5, challenging White's central control and preparing for possible pawn breaks in the center.
You: What is this opening called?
Llama: This is the King's Pawn Opening, specifically the King's Pawn Game or the Open Game after 1.e4.
You: Was the move made by the black player the best possible move? Or do you think he made a mistake?
Llama: The move 2... d5 is considered a sharp and aggressive response by Black, aiming to challenge
White's central control. However, it can also be seen as somewhat provocative and may lead to an
imbalanced position where White has more space but Black gains rapid development. It's not necessarily
a mistake, but rather an attempt to take the initiative and put pressure on White's position.
```

```
rnbqkbnr/ppp2ppp/8/3pp3/4P3/5N2/PPPP1PPP/RNBQKB1R w KQkq - 0 3

1. e4 e5 2. Nf3 d5
```

Fig. 15. Game discussion



```
ANALYSIS (halt?) 83.5M nodes, 2m 3s (N/s: 678k)

focus? 24.2% e4 Qe2 Nf6 d3 Qxd5 Nc3 Bb4 Nd2 Bxc3 bxc3 Qc6 Bb2 Be6 dxe4 Nbd7 Qb5 Qxb5 Bxb5 a6 Be2 Nc5
f3 Na4 O-O-O Nxb2 Kxb2 h5 h4 Nd7 Nb3 Rh6 f4 Bg4 Bxg4 hxg4 c4 Nb6 (Depth: 36, WDL: 1 482 517)

focus? 19.1% Qxd5 Nc3 Qc5 d4 exd4 Nxd4 Nf6 Be2 Be7 Be3 Qa5 Qd2 Nc6 Bb5 Bd7 Nb3 Qb4 O-O-O O-O-O a3 Qh4
Qe2 Bd6 h3 Rhe8 g3 Qh5 g4 Qe5 Qc4 Ne4 Nd5 Kb8 Qa4 (Depth: 36, WDL: 1 380 619)

focus? 10.8% Nd7 d4 exd4 Nxd4 Bc5 Nb3 Bd6 Nc3 Ngf6 Nb5 Ne5 Be2 O-O O-O Re8 c4 Bf8 Bg5 a6 Bxf6 gxf6
N5d4 f5 (Depth: 36, WDL: 0 217 783)

focus? 14.4% Nf6 Nxe5 Qxd5 Qe2 Be7 Nc3 Qd6 b3 O-O Bb2 Nc6 Nxc6 Qxc6 O-O-O Re8 Qb5 Bd7 Qxc6 Bxc6 f3
Rad8 Bd3 h5 Rde1 h4 Bb5 Bxb5 Nxb5 Nd5 Nd4 g6 g3 Bf6 a3 Ne7 Nb5 Bxb2+ Kxb2 Rxd2 Nxc7 (Depth: 35, WDL: 1
```

Chat with Llama!

```
Llama: With 1... e5 being one of the most common responses to 1.e4, Black can also consider 1... e6 or
1... c5 as viable alternatives. However, if I had to pick a popular and solid choice, it would be 1...
e5, challenging White's central control and preparing for possible pawn breaks in the center.
You: What is this opening called?
Llama: This is the King's Pawn Opening, specifically the King's Pawn Game or the Open Game after 1.e4.
You: Was the move made by the black player the best possible move? Or do you think he made a mistake?
Llama: The move 2... d5 is considered a sharp and aggressive response by Black, aiming to challenge
White's central control. However, it can also be seen as somewhat provocative and may lead to an
imbalanced position where White has more space but Black gains rapid development. It's not necessarily
a mistake, but rather an attempt to take the initiative and put pressure on White's position.
You: What is capital of France?
Llama: I'm afraid that's not related to chess! Would you like me to comment on the position after 1.e4
e5 2.Nf3 d5 3.exd5 instead?
```

```
rnbqkbnr/ppp2ppp/8/3Pp3/8/5N2/PPPP1PPP/RNBQKB1R b KQkq - 0 3

1. e4 e5 2. Nf3 d5 3. exd5
```

Fig. 16. Llama's response to non-chess question

6 Conclusion

We successfully created a software integrated with an AI chat bot capable of helping its users to discuss and better understand game strategies during the game play. The LLM, Llama 3.1 model[Grattafiori et al. 2024] that we've used in our project was able to respond to chess-related queries and was in concurrence with the Stockfish[Stockfish Developers 2025] best move predictions. We believe that this software will be able to help people who are new to chess learn at a faster pace than the conventional approach of book-based or chess lessons offered by online services such as Chess.com [Chess.com Developers 2025]. We also feel that this will beneficial for people who won't be able to afford in-person chess lessons or people who have time constraints to attend such classes. At this point, the software requires Ollama[git [n. d.]a] server to be running locally and needs to be set up as a prerequisite for executing the main software. In future, this step could be avoided by hosting the model in a cloud-based server so, that the users need to only set up the main software.

7 Contribution Table

All the members equally contributed to this report. The following table 1 summarizes what each of the members contributed in addition to this report.

Member	Contribution
Vivek Viswam R. V.	System design, Nibbler, Stockfish, Fine-tuning and Ollama set up
Sheethal Urs	Llama and few other model comparison, Nibbler UI update assistance and assisted in fine-tuning step
Sai Mokshith Goud Palle	Studied Chess rating system and past works, tested queries

Table 1. Member Contributions

References

[n. d.]a. GitHub - ollama/ollama: Get up and running with Llama 3.3, DeepSeek-R1, Phi-4, Gemma 3, Mistral Small 3.1 and other large language models. — github.com. https://github.com/ollama/ollama. [Accessed 04-2025].

[n. d.]b. GitHub - rooklift/nibbler: Chess analysis GUI for UCI engines, with extra features for Leela (Lc0) in particular. — github.com. https://github.com/rooklift/nibbler. [Accessed 03-17-2025].

[n. d.]. Node.js — Run JavaScript Everywhere — nodejs.org. https://nodejs.org/en. [Accessed March 2025].

Anthropic AI. 2023a. Claude 3: AI Model for Conversational Intelligence. https://www.anthropic.com/ Accessed: March 2025.

Mistral AI. 2023b. Mistral: Open-Weight Large Language Model. https://mistral.ai/ Accessed: March 2025.

Guillermo Campitelli and Fernand Gobet. 2008. The role of practice in chess: A longitudinal study. *Learning and individual differences* 18, 4 (2008), 446–458.

Chess.com Developers. 2025. Chess.com - Play Chess Online - Free Games — chess.com. https://www.chess.com/. [Accessed February 2025].

Arena Chess GUI Developers. 2023a. Arena Chess GUI: Free Graphical User Interface for Chess Engines. http://www.playwitharena.de/ Accessed: March 2025.

Berserk Chess Developers. 2023b. Berserk: A Lightweight Neural Network Chess Engine. https://github.com/jhonnold/berserk Accessed: March 2025.

Cute Chess Developers. 2023c. Cute Chess: A Simple and Powerful Chess GUI. https://github.com/cutechess/cutechess Accessed: March 2025.

Fairy-Stockfish Developers. 2023d. Fairy-Stockfish: Open-source Chess Engine for Variants. https://github.com/ianfab/Fairy-Stockfish Accessed: March 2025.

Lucas Chess Developers. 2023e. Lucas Chess: Chess Training and Game Playing Interface. https://lucaschess.pythonanywhere.com/ Accessed: March 2025.

Leela Chess Zero Developers. 2023f. Leela Chess Zero (Lc0): Neural Network Chess Engine. https://github.com/LeelaChessZero/lc0 Accessed: March 2025.

Marvin Developers. 2023g. Marvin: A Strong Open-source Chess Engine. https://github.com/bmdanielsson/marvin-chess Accessed: March 2025.

Nibbler Developers. 2023h. Nibbler: A Graphical User Interface for Leela Chess Zero and Other Engines. https://github.com/fohristiwhirl/nibbler Accessed: March 2025.

Qian Dong and Rong Miao. 2020. A comparative study of chess online educational products. In *Blended Learning. Education in a Smart Learning Environment: 13th International Conference, ICBL 2020, Bangkok, Thailand, August 24–27, 2020, Proceedings 13*. Springer, 101–113.

Daniel Dugovic. 2023. Schooner: A Multi-threaded Chess Engine in Rust. https://github.com/ddugovic/Schooner Accessed: March 2025.

Fiverr. 2025. 24 Best Chess Tutor Services To Buy Online | Fiverr — fiverr.com. https://www.fiverr.com/gigs/chess-tutor. [Accessed 25-04-2025].

Fernand Gobet and Guillermo Campitelli. 2006. *The Mechanisms of Skill Acquisition in Chess: The Influence of Practice and Memory*. Psychology Press.

Andy Grant. 2023. Ethereal: A Handcrafted Evaluation Chess Engine. https://github.com/AndyGrant/Ethereal Accessed: March 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

Demis Hassabis. 2017. Artificial intelligence: chess match of the century.

Hui He et al. 2019. Neural Chess: A Framework for Chess Game Analysis Using Deep Learning. *IEEE Transactions on Neural Networks* (2019).

K. Knight and T. Riddle. 2018. Chess as an Interactive Learning Tool. *Cognitive Science* (2018).

X. Li and Y. Zhang. 2018. Chessbot: A Chess-playing Robot. *Robotics and Automation Letters* (2018).

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved Baselines with Visual Instruction Tuning.

Reid McIlroy-Young, Siddhartha Sen, and Jon Kleinberg. 2020. Maia: A Human-like Chess Engine. *arXiv preprint arXiv:2006.01855* (2020).

Mitchell J. 2025. Chess Game Dataset (Lichess) — kaggle.com. https://kaggle.com/datasets/datasnaek/chess. [Accessed February 2025].

H. Müller et al. 2009. Chess Engine Tuning. *Computers Chess Journal* (2009).

OpenAI. 2023. GPT-4: OpenAI's Multimodal Large Language Model. https://openai.com/research/gpt-4 Accessed: March 2025.

D. Sadikov et al. 2006. A Chess Tutor System for Teaching Chess Endgames. *International Conference on Artificial Intelligence* (2006).

Julian Schrittwieser, Ioannis Antonoglou, Karen Simonyan, et al. 2020. Mastering Chess, Shogi and Go without Rules. *Nature* 588 (2020), 502–506.

David Silver, Julian Schrittwieser, Karen Simonyan, et al. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *Nature* 550 (2017), 354–359.

Stockfish Developers. 2025. *Stockfish Chess Engine*. https://stockfishchess.org/ Accessed February 2025.

Y. J. Tee et al. 2021. Enhancing Chess Learning with Chatbots. *Educational AI Review* (2021).

Hugo Touvron et al. 2023. LLaVA: Large Language-Vision Models. *arXiv preprint arXiv:2301.12597* (2023).

Peter Trevithick. 2023. Texel: A Positional Chess Engine with Unique Tuning. https://github.com/peterosterlund2/texel Accessed: March 2025.

Scid vs PC Developers. 2023. Scid vs PC: A Chess Database Application. https://scidvspc.sourceforge.net/ Accessed: March 2025.

Li Wang et al. 2019. Chess Engines: The Evolution of Decision-Making Systems in Chess. *AI Review* (2019).